MRP.py: A Parameterizer of Post-Translationally Modified Residues

Patrick G. Sahrmann,[*,†] Patrick H. Donnan,[†] Kenneth M. Merz, Jr.,[‡] Steven O. Mansoorabadi,[†] and Douglas C. Goodwin[†]

†*Department of Chemistry and Biochemistry, Auburn University, Auburn, Alabama 36849-5312, United States*

‡*Department of Chemistry and the Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan 48824-1312, United States*

E-mail: pgsahrmann@uchicago.edu

Abstract

MRP.py is a Python-based parameterization program for covalently modified amino acid residues for molecular dynamics simulations. Charge derivation is performed via an RESP charge fit, and force constants are obtained through rewriting of either protein or GAFF database parameters. This allows for the description of interfacial interactions between the modifed residue and protein. MRP.py is capable of working with a variety of protein databases. MRP.py's highly general and systematic method of obtaining parameters allows the user to circumvent the process of parameterizing the modified residue-protein interface. Two examples, a covalently bound inhibitor and covalent adduct consisting of modified residues, are provided in the supporting information.

Introduction

Covalently modified amino acids are present in many proteins, from naturally occurring post-translational modifications (PTMs) and adductions to artificially inhibited enzymes via covalent modification. Many PTMs like methylation and hydroxylation are required for enzyme catalytic function. PTMs are often central to metabolic regulatory mechanisms (e.g., phosphorylation, acetylation, etc.)[1] or cellular/membrane targeting (e.g., acylation).[2] There is a growing list of enzymes known to undergo post-translational crosslinking of side chains to produce protein-based cofactors, such as galactose-oxidase, catalase-peroxidase, and cysteine-dioxygenase.[3,4] Furthermore, controlled modification of amino acid residues may provide utility in novel protein engineering and design.[5] Similarly, in the emerging field of peptidomimetics, chemical altering of peptides such as peptide stapling have been shown to confer increased pharmacologic performance.[6] Thus, the understanding of modified residues in enzymology and the variety of roles they play, such as cofactors in catalysis and in the regulation of signal transduction events, has been and continues to be of critical importance.

Molecular dynamics (MD) enables fast simulation of large biomolecules in comparison with a full quantum mechanical (QM) treatment.[7] In MD, the classical dynamics of atoms is replicated computationally through the use of force fields.[8] Force fields seek to accurately describe the potential energy surface (PES) of a molecule through bonding and nonbonding terms. An example force field is given below:

$$\begin{aligned} U = &\sum_{\text{bonds}} k_b(b-b_0)^2 + \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 + \sum_{\text{torsions}} \sum_{n} \frac{1}{2} V_n[1+\cos(n\omega-\gamma)] \\ &+ \sum_{\text{impropers}} k_\omega(\omega-\omega_0)^2 + \sum_{\text{nonbonded}} \Big(\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} + \epsilon_{ij} \Big[\Big(\frac{r_{ij}}{R_{ij}}\Big)^{12} - 2\Big(\frac{r_{ij}}{R_{ij}}\Big)^{6} \Big] \Big) \end{aligned} \tag{1}$$

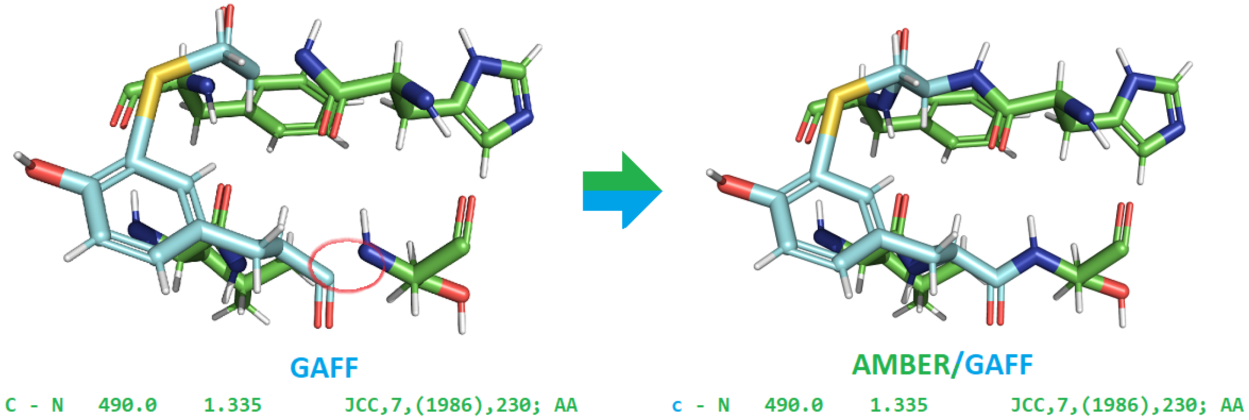
The dynamics of bond lengths and bond angles are encapsulated by harmonic equations for potential energy terms, with b and θ representing the bond lengths and angles, k_b, k_θ, representing the bond and angle spring constants, and b_0 and θ_0 representing the zero-energy bond and angle values. The energy due to torsion constraints is represented by a Fourier expansion term, where V_n is the torsion barrier, n is the periodicity, ω is the torsion angle, and γ is the phase. Improper torsions are also included as a harmonic restraint, with k_ω and ω_0 representing the corresponding force constant and zero-energy angle, respectively. Together, these three terms represent all bonding interactions within a molecule. The last two terms correspond to the nonbonding interactions between atoms and include Coulombic interactions and the Lennard-Jones potential to account for charged species and van der Waals terms, respectively. Partial charges of atoms are represented by q_i and q_j, while R_{ij} and ϵ_{ij} correspond to the well minima and depth, respectively.

Accurate molecular modeling relies on accurate deduction of force constants for each term in the force field. Several methods are available to determine these force constants, such as through experimental (e.g., NMR, normal-mode analysis),[9,10] or purely theoretical means, namely via Hessian matrix calculations from a QM-derived PES.[11] Due to the challenge of the former as well as the ease and high degree of accuracy of the latter, theoretically determined force constants are often preferred. For large biomolecules such as proteins, force field databases (e.g., ff14SB) are available, which encompass the majority of bonding and nonbonding interactions in amino acid residues.[12]

AMBER[13] is a software package for the simulation of biomolecules via molecular dynamics. AMBER features a variety of tools to handle ligand parameterization, such as antechamber[14] and MCPB.py[15] for organic and organometallic ligands, respectively. Ligand parameterization can be partitioned primarily into two parts, charge derivation and force constant assignment. Most charge derivations, including those for the ff14SB database, are performed via a restrained electrostatic potential (RESP) fit.[16,17] Variability in RESP fits stem from approaches to the QM calculation and selection of restrained atoms. Force constant assignment, however, provides more options. Force constant estimation of nonstandard parameters (parameters not documented in force field databases), such as that featured in the parmchk program in antechamber or via *ab initio* force constant derivation (e.g., using CartHess2FC.py) provide alternatives to database force constants.[14]

Additionally, the general AMBER force field (GAFF) is capable of parameterizing the majority of organic molecules, namely molecules that are comprised of C, N, O, H, S, P, F, Cl, Br and I.[18] AmberTools recognizes atoms as belonging to either the AMBER or GAFF atom type by upper or lower case designation, respectively. AMBER atom bonding parameters are located in protein databases such as ff14SB,[7] whereas GAFF atom bonding parameters are found in the GAFF database.[18]

Modeling of post-translationally modified proteins presents a unique challenge in that designating the modified residue as either an amino acid via the AMBER atom type or an organic molecule via the GAFF atom type leads to different complications when parameterization is attempted. Similar issues have arisen in other molecular dynamics software packages such as GROMACS,[19] for which the Vienna-PTM Web Server[20] has emerged as an automated PTM parameterizer. The residuegen program in the antechamber suite[14] enables parameterization of modified residues and is capable of handling multiple residue conformations, whereas MRP.py should be used when only one residue conformation is needed. To our knowledge, MRP.py is the first application to streamline and generalize the complete parameterization of proteins with PTMs using the AMBER and GAFF force fields. AMBER atom types are convenient to use to connect the modified residue to the rest of the protein; however, bonding not classified by ff14SB, which is common in PTM residues, will inevitably leave missing parameters. Conversely, GAFF atom types enable coverage of the majority of bonding interactions within the modified residue, but fail to connect the modified residue to the rest of the protein via the main chain, as there is no direct communication between AMBER and GAFF atom types. To circumvent this issue, the program developed herein treats the modified residue using GAFF atom types, and writes parameters describing the connection of the protein to the modified residue by finding the corresponding ff14SB or GAFF parameter and rewriting it as a connection between an AMBER atom and GAFF atom.



```
C - N   490.0    1.335       JCC,7,(1986),230; AA
```

```
c - N   490.0    1.335       JCC,7,(1986),230; AA
```

Figure 1: Conversion of a ff14SB parameter to a hybrid AMBER/GAFF parameter to describe bonding between a modified residue and a standard residue in the enzyme cysteine dioxygenase (CDO). CDO is known to have a post-translationally formed adduct between a tyrosine and a cysteine side chain.[21] Modified residue carbons are blue and standard residue carbons are green.

Methodology

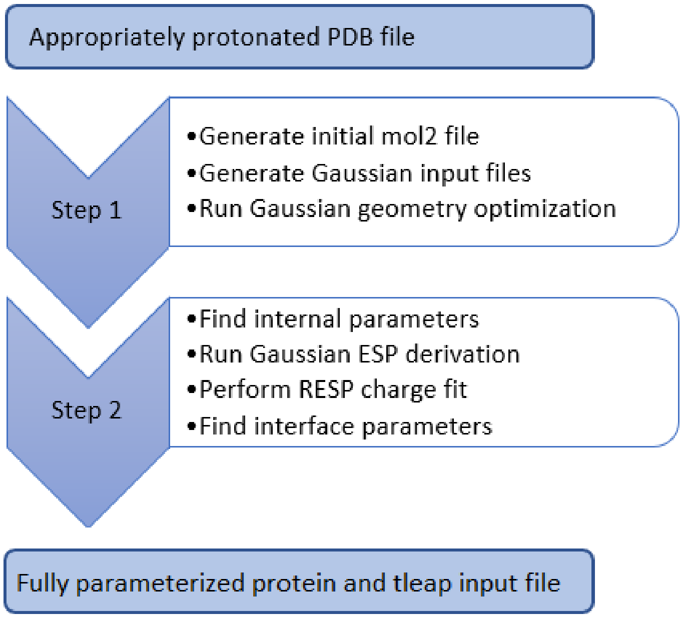
The input file of MRP.py streamlines the parameterization process. The type of modified residue will determine the amount of input variables that must be adressed within the input file, as demonstrated in the supporting information examples. ACE (CO-CH_3) and NME (NH-CH_3) functional groups are used to cap the ends of each modified residue. Once capped, a two step RESP charge fit, analogous to that found in the antechamber[22] procedure is employed to assign charges to each atom within the residue. The first step consists of a charge fit with no restraints on the residue, while the second step involves isomerizing the charges on methyl and methylene hydrogens. In both cases, the charges of the ACE and NME groups are fixed such that each has a total charge of zero. Following the assignment of charges, MRP.py renames the residue to distinguish it from standard residue types recognized by AMBER and creates a corresponding mol2 file for the newly named residue. MRP.py then loads the PDB file into tleap to find missing parameters, which consist of AMBER and GAFF atom type bonding interactions. MRP.py employs a conversion of the GAFF atom types of the residue to AMBER atom types and searches for the corresponding parameters in the ff14SB database, which describes bonding interactions of AMBER atom types, and returns the appropriate missing parameters. This conversion process is illustrated in Figure 1. If parameters are still missing, the inverse conversion from AMBER to GAFF atom types is performed and parameters are searched for in the GAFF database. Fully parameterized, the PDB can then be loaded into tleap with a ready-made input file written by MRP.py. It is important to note that because MRP.py employs GAFF for describing bonding interactions within the modified residue, MRP.py is capable of parameterizing a wide range of modified residues. A glossary of input keywords and their meanings are illustrated in Table 1.

Table 1: Glossary of keywords necessary when using MRP.py.

pdb_name	PDB filename to extract modified residue
project_name	Names of files pertinent to modified residue will begin with project name
residue_ids	Numerical index of modified residues in pdb file
adduct_connection_atoms	Numerical index of atoms that bridge the modified residue and other molecule/residue
adduct_connection_ids	GAFF atom types of atoms that bridge the modified residue and other molecule/residue
charge	Total charge of modified residue and bound species
multiplicity	Spin multiplicity of modified residue and bound species
water_model	Water type for protein to be solvated in

The code can be obtained at GitHub (https://github.com/pgsahrmann/mrp). Two examples are provided in the Supporting Information demonstrating the capability of MRP.py to parameterize covalently bound inhbitors and covalent adducts. The first is human microphage inhibitory factor exhibiting a proline covalently bound to phenethylisothiocyanate.[23–25] The second is catalase-peroxidase (KatG) from the bacterium *Burkholderia pseudomallei*, an enzyme which bears a three amino-acid covalent adduct.[26–29] Minimization, heating, equilibration and MD simulations were conducted with the GPU version of the Amber 2016 package for both examples.[30–32] Constrained hydrogen dynamics were employed, and a different water model was used for solvation in each example.[33–35]

Workflow. MRP.py requires little prior knowledge of parameterization and charge deriva-



Appropriately protonated PDB file
Step 1
•Generate initial mol2 file
•Generate Gaussian input files
•Run Gaussian geometry optimization
Step 2
•Find internal parameters
•Run Gaussian ESP derivation
•Perform RESP charge fit
•Find interface parameters
Fully parameterized protein and tleap input file

Scheme 1: Workflow of MRP.py.

tion, as well as minimal user interference, to successfully operate. Scheme 1 illustrates the workflow of MRP.py. MRP.py is designed to work with AmberTools and the Gaussian software package to handle molecular modeling and QM calculations, respectively.[36] AmberTools15[13] or higher should be used in conjuction with MRP.py, as data files and the resp program from AmberTools15 are required. The ff14SB database is the default protein database used by MRP.py; however, MRP.py is also compatible with ff94[37] and ff99.[38] MRP.py is structured around the Python Metal Site Modeling Toolbox (pyMSMT), and incorporates the molecule class from pyMSMT. Both the protein and the modified residue(s) should be appropriately protonated. This can be accomplished through available software such as ProteinPrepare[39] and H++[40] . MRP.py first caps the modified residue(s) present and creates an initial mol2 file and Gaussian input file for geometry optimization and electrostatic surface potential (ESP) generation. The initial mol2 file allows for the user to determine whether to keep the designated GAFF atom types chosen by antechamber for the modified residue or to designate their own for the purposes of later parameterization. Geometry optimization of the capped modified residue is then performed. In the second step, ESP generation and the subsequent RESP charge fit is performed. Next, any missing GAFF parameters describing the bonding interactions within the modified residue are found and written to an frcmod file via parmchk. Mol2 files are then written for each modified residue containing the RESP fitted charges, and interface parameters (missing parameters describing the connection of the modified residue to the protein) are written to another frcmod file. The user is then free to edit an autogenerated tleap input file before generation of the solvated protein topology and coordinate files. MRP.py thus not only parameterizes modified residues but automates protein preparation and incorporation of related parameter files through this tleap input file.

Conclusion

The prevalence of modified residues in enzymology warrants a systematic and simple method of parameterization for molecular modeling of these species. In light of recent software such as MCPB.py to streamline the parameterization process of nontrivial ligands, MRP.py provides a simple procedure to aid in the simulation of biomolecules. MRP.py is capable of parameterizing modified residues in a variety of biological contexts, such as covalently bound inhibitors or covalently linked residues. MRP.py is intended to efficiently streamline this process with minimal user interference. MRP.py is written to work with the molecular modeling software package AmberTools and the QM software Gaussian.

Associated Content

Supporting Information

Two detailed examples using AMBER 2016 and Gaussian 16 (.docx). The MRP.py input file, protonated PDB file, mol2 files, Gaussian .com and .log files, frcmod files, and tleap input file for each of the two examples (.zip).

Author Information

Corresponding Author

*Patrick G. Sahrmann, E-mail: pgsahrmann@uchicago.edu.

Notes

The authors declare no competing financial interest.

Acknowledgements

This work was supported in part by funds from a National Science Foundation Grant awarded to DCG (MCB-1616059) and a CAREER award (CHE-1555138) to SOM. PGS was supported in part as a G. Lynn and Cheryl Marks Family Research Scholar. PHD acknowledges support from the National Science Foundation Established Program to Stimulate Competitive Research. This work was supported in part by a grant of high-performance computing resources and technical support from the Alabama Supercomputer Authority. The authors thank Jessica R. Krewall for helpful discussions.

References

(1) Deribe, Y. L.; Pawson, T.; Dikic, I. Post-Translational Modifications in Signal Integration. *Nat. Struct. Mol. Biol.* **2010**, *17*, 666–672.

(2) Drazic, A.; Myklebust, L. M.; Ree, R.; Arnesen, T. The World of Protein Acetylation. *Biochim. Biophys. Acta, Proteins Proteomics* **2016**, *1864*, 1372–1401.

(3) Njuma, O. J.; Ndontsa, E. N.; Goodwin, D. C. Catalase in Peroxidase Clothing: Interdependent Cooperation of Two Cofactors in the Catalytic Versatility of KatG. *Arch. Biochem. Biophys* **2014**, *544*, 27–39.

(4) Davidson, V. L. Protein-Derived Cofactors Revisited: Empowering Amino Acid Residues with New Functions. *Biochemistry* **2018**, *57*, 3115–3125.

(5) Karle, I. L.; Balaji Rao, R.; Prasad, S.; Kaul, R.; Balaram, P. Nonstandard Amino Acids in Conformational Design of Peptides. Helical Structures in Crystals of 5-10 Residue Peptides Containing Dipropylglycine and Dibutylglycine. *J. Am. Chem. Soc.* **1994**, *116*, 10355–10361.

(6) Verdine, G. L.; Hilinski, G. J. *Methods in Enzymology*; Elsevier, 2012; pp 3–33.

(7) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696–3713.

(8) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields. *Nat. Commun.* **2018**, *9*, 3887.

(9) Tirion, M. M. Large Amplitude Elastic Motions in Proteins from a Single-Parameter, Atomic Analysis. *Phys. Rev. Lett.* **1996**, *77*, 1905–1908.

(10) Langer, J.; Schrader, B.; Bastian, V.; Jacob, E. Infrared Spectra and Force Constants of Urea in the Gaseous Phase. *Fresenius J. Anal. Chem.* **1995**, *352*, 489–495.

(11) Seminario, J. M. Calculation of Intramolecular Force Fields from Second-Derivative Tensors. *Int. J. Quantum Chem.* **1996**, *60*, 1271–1277.

(12) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins* **2006**, *65*, 712–725.

(13) Case, D. A. e. a. AMBER. **2016**,

(14) Wang, J.; Wang, W.; A. Kollman, P.; Case, D. A. ANTECHAMBER: An accessory Software Package for Molecular Mechanical Calculations. *J. Chem. Inf. Comput. Sci.* **2001**, *222*, 403.

(15) Li, P.; Merz, K. M. MCPB.py: A Python Based Metal Center Parameter Builder. *J. Chem. Inf. Model.* **2016**, *56*, 599–604.

(16) Aronica, P. G. A.; Fox, S. J.; Verma, C. S. Comparison of Charge Derivation Methods Applied to Amino Acid Parameterization. *ACS Omega* **2018**, *3*, 4664–4673.

(17) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Kollman, P. A. Application of RESP Charges to Calculate Conformational Energies, Hydrogen Bond Energies, and Free Energies of Solvation. *J. Am. Chem. Soc.* **1993**, *115*, 9620–9631.

(18) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J. Comput. Chem.* **2004**, *25*, 1157–1174.

(19) Spoel, D. V. D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J. Comput. Chem.* **2005**, *26*, 1701–1718.

(20) Margreitter, C.; Petrov, D.; Zagrovic, B. Vienna-PTM Web Server: A Toolkit for MD Simulations of Protein Post-Translational Modifications. *Nucleic Acids Res.* **2013**, *41*, 422–426.

(21) Ito, N.; Phillips, S. E. V.; Stevens, C.; Ogel, Z. B.; McPherson, M. J.; Keen, J. N.; Yadav, K. D. S.; Knowles, P. F. Novel Thioether Bond Revealed by a 1.7 Å Crystal Structure of Galactose Oxidase. *Nature* **1991**, *350*, 87–90.

(22) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic Atom Type and Bond Type Perception in Molecular Mechanical Calculations. *J. Mol. Graph. Model.* **2006**, *25*, 247–260.

(23) Schrader, J.; Deuster, O.; Rinn, B.; Schulz, M.; Kautz, A.; Dodel, R.; Meyer, B.; Al-Abed, Y.; Balakrishnan, K.; Reese, J. P.; Bacher, M. Restoration of Contact Inhibition in Human Glioblastoma Cell Lines after MIF Knockdown. *BMC Cancer* **2009**, *9*, 464.

(24) Cheung, K. L.; Kong, A.-N. Molecular Targets of Dietary Phenethyl Isothiocyanate and Sulforaphane for Cancer Chemoprevention. *AAPS J* **2009**, *12*, 87–97.

(25) Crichlow, G. V.; Fan, C.; Keeler, C.; Hodsdon, M.; Lolis, E. J. Structural Interactions Dictate the Kinetics of Macrophage Migration Inhibitory Factor Inhibition by Different Cancer-Preventive Isothiocyanates. *Biochemistry* **2012**, *51*, 7506–7514.

(26) Zámocký, M.; Obinger, C. *Biocatalysis Based on Heme Peroxidases*; Springer Berlin Heidelberg, 2010; pp 7–35.

(27) Zhang, Y.; Heym, B.; Allen, B.; Young, D.; Cole, S. The Catalase—Peroxidase Gene and Isoniazid Resistance of Mycobacterium tuberculosis. *Nature* **1992**, *358*, 591–593.

(28) Kudalkar, S. N.; Campbell, R. A.; Li, Y.; Varnado, C. L.; Prescott, C.; Goodwin, D. C. Enhancing the Peroxidatic Activity of KatG by Deletion Mutagenesis. *J. Inorg. Biochem.* **2012**, *116*, 106–115.

(29) Loewen, P. C.; Carpena, X.; Vidossich, P.; Fita, I.; Rovira, C. An Ionizable Active-Site Tryptophan Imparts Catalase Activity to a Peroxidase Core. *J. Am. Chem. Soc* **2014**, *136*, 7249–7252.

(30) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular Dynamics with Coupling to an External Bath. *J. Chem. Phys.* **1984**, *81*, 3684–3690.

(31) Götz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; Grand, S. L.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *J. Chem. Theory Comput.* **2012**, *8*, 1542–1555.

(32) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Grand, S. L.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9*, 3878–3888.

(33) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes. *J. Comput. Phys.* **1977**, *23*, 327–341.

(34) Izadi, S.; Anandakrishnan, R.; Onufriev, A. V. Building Water Models: A Different Approach. *J. Phys. Chem. Lett.* **2014**, *5*, 3863–3871.

(35) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* **1983**, *79*, 926–935.

(36) M. J. Frisch, e. a. Gaussian 16 Revision B.01. Gaussian Inc.: Wallingford CT, 2016.

(37) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **1995**, *117*, 5179–5197.

(38) Wang, J.; Cieplak, P.; Kollman, P. A. How Well Does a Restrained Electrostatic Potential (RESP) Model Perform in Calculating Conformational Energies of Organic and Biological Molecules? *J. Comput. Chem.* **2000**, *21*, 1049–1074.

(39) Martínez-Rosell, G.; Giorgino, T.; Fabritiis, G. D. PlayMolecule ProteinPrepare: A Web Application for Protein Preparation for Molecular Dynamics Simulations. *J. Chem. Inf. Model.* **2017**, *57*, 1511–1516.

(40) Anandakrishnan, R.; Aguilar, B.; Onufriev, A. V. H++ 3.0: Automating pK Prediction and the Preparation of Biomolecular Structures for Atomistic Molecular Modeling and Simulations. *Nucleic Acids Res.* **2012**, *40*, 537–541.

Graphical TOC Entry